**README**

**Deep learning final project - project\_307923052\_302636675**

1. Downlad the code from the git site.

2. Download Dataset

Since the illustration dataset can't be publicly avalable (legal rights) we will not provide the drive link for the dataset here.

We shared a folder named "**project\_307923052\_302636675"** with barakhadad@mail.tau.ac.il (if you encounter any problems please contact us and we will respond immediately

noabarzilay11@gmail.com).

**StarGAN framework-**

In order to be able to run the code with the illustration dataset download go to:

the folder project\_307923052\_302636675 -> DATASET -> StarGanDataset

And download the illustrations.zip file and extract it in the following path:

your\_path\_to\_our\_project\DeepLearningProject\stargan\data\Extract the folder here

**MUNIT framework-**

In order to be able to run the code with the illustration dataset download go to:

the folder project\_307923052\_302636675 -> DATASET -> MUNITDataset

And download the illustrations2landscapes.zip file and extract it in the following path:

your\_path\_to\_our\_project\DeepLearningProject\MUNIT\datasets\Extract the folder here

3. **Training**

in order to train the different genrators fro scratch-

**StarGAN framework-**

In the report we described several models, we are going to elaborate here how to train each model:

MillGAN -

python main.py --mode train --num\_domains 2 --w\_hpf 0 --lambda\_reg 1 --lambda\_sty 1 --lambda\_ds 1 --lambda\_cyc 1 --train\_img\_dir data\illustrations\train --val\_img\_dir data\illustrations\val --vgg\_w 1

Model A- Baseline StarGan -

python main.py --mode train --num\_domains 2 --w\_hpf 0 --lambda\_reg 1 --lambda\_sty 1 --lambda\_ds 1 --lambda\_cyc 1 --train\_img\_dir data\illustrations\train --val\_img\_dir data\illustrations\val --use\_star\_gen 1

Model B-

python main.py --mode train --num\_domains 2 --w\_hpf 0 --lambda\_reg 1 --lambda\_sty 1 --lambda\_ds 1 --lambda\_cyc 1 --train\_img\_dir data\illustrations\train --val\_img\_dir data\illustrations\val --use\_residual\_upsample 0

Model D-

python main.py --mode train --num\_domains 2 --w\_hpf 0 --lambda\_reg 1 --lambda\_sty 1 --lambda\_ds 1 --lambda\_cyc 1 --train\_img\_dir data\illustrations\train --val\_img\_dir data\illustrations\val --loss\_sacl 1

Model E

python main.py --mode train --num\_domains 2 --w\_hpf 0 --lambda\_reg 1 --lambda\_sty 1 --lambda\_ds 1 --lambda\_cyc 1 --train\_img\_dir data\illustrations\train --val\_img\_dir data\illustrations\val --vgg\_w 1 --loss\_sacl 1

**MUNIT framework-**

In the report we described several models, we are going to elaborate here how to train each model:

Model A - baseline-

Go to "illustrations2landscapes\_folder.yaml" in MUNIT\configs-

Chane in the file the following-

ganilla\_gen: False

Model B-

Chane in the file the following-

use\_style\_enc\_simple: True

Model C-

Chane in the file the following-

use\_style\_enc\_simple: False

Model E-

Chane in the file the following-

use\_patch\_gan: False

Model F-

Chane in the file the following-

style\_dim: 16

For all Run the following command -

**python train.py --config configs/illustrations2landscapes\_folder.yaml**

4. **Evaluating pretrained models -**

**StarGAN framework-**

In order to download pretrained models go to:

the folder project\_307923052\_302636675 -> TrainedGenerators-> StarGan

Pick the pretrained model you wish to evaluate and enter the folder (for example "MillGAN").

The dorralated pretrained models according to the report-

Model A - exprOriginal

Model B- exprGanilla\_AdainNoResidualinup

Model C- exprGanilla\_AdainResBlocksinUp

Model D- exprGanilla\_AdainResBlocksinUp\_SACL

Model E- exprGanilla\_AdainResBlocksinUp\_VGGContentLoss\_SACL

MillGAN - it's simply MillGAN :)

All other models in the drive appears in the Appendix of the report.

copy the content of the checkpoints folder from the model you pick to :

your\_path\_to\_our\_project\DeepLearningProject\stargan\expr\checkpoints

Rum the following command -

**IMPORTANT** - You need to add the additional flags presented in the former section to the command line here as well (for example for evaluating the baselime model you shuld run the commands with --use\_star\_gen 1)

For Generate images:

--mode eval --num\_domains 2 --w\_hpf 0 --resume\_iter 100000 --train\_img\_dir data\illustrations\train --val\_img\_dir data\illustrations\val --checkpoint\_dir expr\checkpoints\ --eval\_dir expr\eval

For generate images from refences illustrations:

Create a folder with the following path:

assets\representative\illustrations\src

Drop there all the **natural** images you wish to transfer to illustrations.

Create another folder with the folwing path:

assets\representative\illustrations\ref

Drop there all of the **illustration** images you wish the generator will extract the style code from.

Run the following command and you will get an image simillar to Figure 9 from report:

--mode sample --num\_domains 2 --resume\_iter 100000 --w\_hpf 0 --checkpoint\_dir expr\checkpoints\ --result\_dir expr\results\ --src\_dir assets\representative\illustrations\src --ref\_dir assets\representative\illustrations\ref

**MUNIT framework-**

In order to download pretrained models go to:

the folder project\_307923052\_302636675 -> TrainedGenerators-> MUNIT

Download the models direcory and copy it to:

your\_path\_to\_our\_project\DeepLearningProject\MUNIT\models

Model A - OriginalMUNIT

Model B- allup\_adain\_vgg\_ncyc\_styleEncMunit\_total

Model C- allup\_adain\_vgg\_cyc\_total

Model D- allup\_adain\_vgg\_ncyc\_styleEncMunit5nd\_total

Model E- allup\_adain\_vgg\_ncyc\_styleEncMunit\_total\_MSdis

ModelF - allup\_adain\_vgg\_ncyc\_styleEncMunit\_16styledim\_total

All other models in the drive appears in the Appendix of the report.

**IMPORTANT** - Each model you wish to test shole be tested with the proper configuration presented in the former section "Training", for example, Model A should contain in the illustrations2landscapes\_folder.yaml file "ganilla\_gen: False".

--config configs/illustrations2landscapes\_folder.yaml --input inputs/Input\_image\_you\_chose --output\_folder results/model\_results --checkpoint models/Pre-trained\_model\_you\_chose.pt --a2b 0